

19) **BUNDESREPUBLIK
DEUTSCHLAND**



**DEUTSCHES
PATENT- UND
MARKENAMT**

12) **Übersetzung der
europäischen Patentschrift**

51) Int. Cl.⁷:
G 10 L 13/00

97) **EP 0 691 023 B 1**

10) **DE 694 20 955 T 2**

21) Deutsches Aktenzeichen:	694 20 955.4
66) PCT-Aktenzeichen:	PCT/GB94/00430
96) Europäisches Aktenzeichen:	94 908 433.9
87) PCT-Veröffentlichungs-Nr.:	WO 94/23423
86) PCT-Anmeldetag:	7. 3. 1994
87) Veröffentlichungstag der PCT-Anmeldung:	13. 10. 1994
97) Erstveröffentlichung durch das EPA:	10. 1. 1996
97) Veröffentlichungstag der Patenterteilung beim EPA:	29. 9. 1999
47) Veröffentlichungstag im Patentblatt:	13. 7. 2000

30) Unionspriorität:
93302383 26. 03. 1993 EP

73) Patentinhaber:
British Telecommunications p.l.c., London, GB

74) Vertreter:
Beetz und Kollegen, 80538 München

84) Benannte Vertragsstaaten:
BE, CH, DE, DK, ES, FR, GB, IT, LI, NL, SE

72) Erfinder:
GAVED, Margaret, Ipswich, Suffolk IP4 4DU, GB;
HAWKEY, James, Huang Pu District, Shanghai,
China Post, CN

54) **UMWANDLUNG VON TEXT IN SIGNALFORMEN**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

DE 694 20 955 T 2

DE 694 20 955 T 2

BEST AVAILABLE COPY

13.12.99

EP 0 691 023

Die Erfindung betrifft ein Verfahren und eine Vorrichtung zur Umwandlung eines Textes in eine Wellenform. Insbesondere betrifft sie die Erzeugung einer Ausgabe in Form einer akustischen Welle, nämlich synthetischer Sprache, aus einer Eingabe in Form von Signalen, die einen herkömmlichen Text darstellen.

Diese Gesamtumwandlung ist sehr kompliziert und wird manchmal in mehreren Modulen durchgeführt, bei denen der Ausgang eines Moduls den Eingang für das nächste Modul bildet. Das erste Modul empfängt Signale, die einen herkömmlichen Text darstellen, und das Endmodul erzeugt an seinem Ausgang eine synthetische Sprache. Diese synthetische Sprache kann eine digitale Darstellung der Wellenform, der eine herkömmliche digital-analoge Umwandlung folgt, um eine akustische Ausgabe zu erzeugen. In vielen Fällen ist es erwünscht, die akustische Ausgabe über ein Telefonsystem zu realisieren. In diesem Fall kann es geeignet sein, die Digital-analog-Umwandlung nach dem Senden durchzuführen, so daß das Senden in digitaler Form stattfindet.

Der modulare Aufbau hat Vorteile; z. B. wird jedes Modul separat entwickelt und jedes beliebige der Module kann ersetzt oder geändert werden, um die Flexibilität, Verbesserungen oder eine Anpassung an geänderte Umstände zu erreichen.

Derartige Verfahren nutzen eine Folge von drei Modulen, nämlich

- (A) Voreditieren,
- (B) Umwandlung von Graphemen in Phoneme, und

(C) Umwandlung von Phonemen in (digitale) Wellenformen.

Nachfolgend werden diese Module kurz beschrieben.

Das Modul (A) empfängt Signale, die einen herkömmlichen Text darstellen, z. B. den Text dieser Beschreibung, und modifiziert ausgewählte Merkmale. Somit kann das Modul (A) bestimmen, welche Nummern verarbeitet werden. Zum Beispiel wird es entscheiden, ob

"1345"

zu

eins, drei, vier, fünf
dreizehn fünfundvierzig oder
eintausenddreihundertfünfundvierzig werden.

Es ist ersichtlich, daß es relativ einfach ist, verschiedene Formen des Moduls (A) darzustellen, wobei jede von ihnen mit den nachfolgenden Modulen kompatibel ist, so daß verschiedene Formen der Ausgabe entstehen.

Das Modul (B) wandelt Grapheme in Phoneme um. Die "Grapheme" sind Datendarstellungen, die den Symbolen des gewöhnlichen Alphabets entsprechen, die auf gewöhnliche Art und Weise verwendet werden. Der Text dieser Beschreibung ist ein gutes Beispiel für die "Grapheme". Es ist ein Problem der synthetischen Sprache, daß die Grapheme nur geringen Bezug zu der Art haben können, wie Wörter ausgesprochen werden, insbesondere in Sprachen wie Englisch. Um Wellenformen zu erzeugen ist es deswegen angebracht, die Grapheme in ein anderes Alphabet umzuwandeln, das in dieser Beschreibung "Phoneme" genannt wird, das eine sehr enge Korrelation mit dem Klang der Wörter aufweist. Mit anderen Worten ist der Zweck des Moduls (B), das Problem zu lösen, daß das gewöhnliche Alphabet nicht phonetisch ist.

Das Modul (C) wandelt die Phoneme in eine digitale Wellenform um, die, wie oben erwähnt wurde, in ein Analogformat und somit in eine akustische Wellenform umgewandelt werden kann.

Die vorliegende Erfindung betrifft ein Verfahren und eine Einrichtung zur Verwendung im Modul (B), wobei dieses Modul jetzt ausführlicher beschrieben wird.

Das Modul (B) verwendet verknüpfte Datenbanken, die durch eine große Anzahl von unabhängigen Einträgen gebildet werden. Jeder Eintrag umfaßt Zugriffsdaten, die in Form von Darstellungen, z.B. Bytes, einer Folge von Graphemen vorhanden sind, und eine Ausgangskette, die Darstellungen, z.B. Bytes, der Phoneme umfaßt die den Graphemen entsprechen, die im Zugriffsbereich enthalten sind. Ein großes Problem der Umwandlung Grapheme/Phoneme besteht in der Größe der Datenbank, die notwendig ist, um mit einer Sprache fertig zu werden. Eine einfache und theoretisch ideale Lösung wäre, eine so große Datenbank vorzusehen, daß sie einen individuellen Eintrag für jedes mögliche Wort in der Sprache einschließlich sämtlicher möglichen Intonationen jedes möglichen Wortes in der Sprache aufweist. Selbstverständlich würde durch Verwendung einer kompletten Datenbank jedes Wort im Eingabetext individuell erkannt und ein hervorragendes Phonemäquivalent ausgegeben. Es ist jedoch ersichtlich, daß es nicht möglich ist, eine derart komplette Datenbank zu verwenden. In erster Linie ist es nicht möglich, jedes Wort in einer Sprache zu erfassen und selbst wenn ein solches Verzeichnis verfügbar wäre, wäre es für Computerzwecke zu umfangreich.

Obwohl die komplette Datenbank nicht realisierbar ist, ist es möglich, eine Datenbank von verwendbarer Größe vorzusehen, die z.B. gemeinsame Wörter und Wörter enthält, deren

Aussprache bezüglich der Rechtsschreibung nicht einfach ist. Eine derartige Datenbank ermöglicht eine hervorragende Grapheme/Phoneme-Umwandlung der darin enthaltenen Wörter, fällt jedoch bei fehlenden Wörtern aus, d.h., erzeugt keine Ausgabe. Bei jeder praktischen Anwendung würde dies eine nicht vertretbar hohe Fehlerquote bedeuten.

Eine andere Möglichkeit verwendet eine Datenbank, in der die Zugriffsdaten kurzen Zeichenfolgen von Graphemen entsprechen, von denen jede Zeichenfolge mit ihrer äquivalenten Zeichenfolge von Phonemen verknüpft ist. Diese Alternative verwendet eine verwaltbare Größe der Datenbank; es hängt jedoch von der Analyse des Eingabetextes ab, ob die darin enthaltenen Zeichenfolgen den Zugriffsdaten in der Datenbank entsprechen. Systeme dieser Art können einen großen Anteil an hervorragenden Aussprachen mit leichten und einzelnen Aussprache Fehlern gewährleisten. Auch ist ein Anteil an Ausfällen vorhanden, bei denen keine Ausgabe erzeugt wird, entweder auf Grund der fehlerhaften Analyse oder weil im Zugriffsbereich der Datenbank eine erforderliche Zeichenfolge von Graphemen fehlt.

Die letzte Möglichkeit ist üblicherweise als ein Standard-Verfahren bekannt, weil sie von n nur dann angewandt wird, wenn bevorzugte Verfahren versagen. Ein Standard-Verfahren besteht üblicherweise in der "Aussprache" der Symbole des Eingabetextes. Da der Bereich der Eingabesymbole nicht nur bekannt, sondern auch begrenzt ist (gewöhnlich weniger als 100 und in vielen Fällen weniger als 50), ist es nicht nur möglich, die Datenbank zu erzeugen, sondern ist auch deren Größe im Vergleich mit der Kapazität von modernen Datenspeichersystemen sehr klein. Deswegen gewährleistet dieses Standard-Verfahren eine Ausgabe selbst dann, wenn diese Ausgabe nicht die beste Lösung darstellt. Beispiele bilden Namen, in denen Initialen verwendet werden, akademische

Grade und Ehrentitel, und Abkürzungen von Einheiten. Es wird geschätzt, daß es unter diesen Umständen üblich ist, die Buchstaben "auszusprechen", wobei das Standard-Verfahren in diesen Fällen die besten Ergebnisse liefert.

Bisher wurden drei verschiedene Strategien zur Umwandlung von Graphemen in Phoneme ermittelt und es ist wichtig, zu begreifen, daß diese Alternativen sich nicht gegenseitig ausschließen. Es ist sogar erwünscht, alle drei Alternativen gemäß einer strikten Prioritätsregel zu verwenden. So wird als erste die "Ganzwort"-Datenbank verwendet und wenn sie eine Ausgabe erzeugt, dann wird diese hervorragend sein. Wenn sie ausfällt, wird das "Analyse"-Verfahren verwendet, das eine kleine, aber annehmbare Anzahl von Aussprachefehlern umfaßt. Schließlich, wenn das "Analyse"-Verfahren versagt, wird die Standard-Option der Aussprache der "Buchstaben" verwendet, die garantiert eine Ausgabe erzeugt. Obwohl diese Lösung nicht ganz zufriedenstellend ist, bietet sie unter Berücksichtigung der oben erwähnten Fälle das beste Ergebnis.

Die vorliegende Erfindung betrifft die mittlere Option von der oben beschriebenen Reihenfolge. Das bedeutet, daß diese Erfindung die Analyse der Datendarstellungen betrifft, die den Eingabetextgraphemen entsprechen, um einen Ausgabesatz von Datendarstellungen zu erzeugen, die dem Eingabetext entsprechende Phoneme sind. Es wird betont, daß das Arbeitsgebiet dieser Erfindung eine vollständige Text-in-Wellenform-Umwandlung ist, wie ausführlicher oben beschrieben wurde. Das bedeutet, daß diese Erfindung eine besondere Komponente des Gesamtsystems betrifft.

Das Dokument von F. F. Lee (Seiten 333 - 338), veröffentlicht in "PROCEEDINGS OF THE SPRING JOINT COMPUTER CONFERENCE" am 30. April 1968 in Atlantic City, NJ, betrifft

eine durch Computer erzeugte Sprache. Das Dokument beschreibt die Aufspaltung von Wörtern in "Morphe". Es wird erwähnt, daß wenn zwei Morphe kombiniert werden, die Änderungen in der Rechtschreibung lediglich im linken Morphen erfolgen, so daß es geeignet ist, ein gedrucktes Wort während des Aufspaltungsprozesses von rechts nach links abzutasten. Die "Morphe" werden als die kleinsten bedeutenden Einheiten in geschriebener Form definiert.

Gemäß der vorliegenden Erfindung wird eine Eingabefolge von Bytes, z.B. Datendarstellungen, die eine Folge von Zeichen darstellen, die aus einem ersten Zeichensatz, wie Graphemen ausgewählt werden, in Unterfolgen aufteilt, zur Umwandlung in eine Ausgabefolge von Bytes, z.B. Datendarstellungen, die eine Folge von Zeichen darstellen, die aus einem zweiten Zeichensatz, wie Phonemen, ausgewählt werden, wobei das Verfahren eine rückläufige Analyse umfaßt und dadurch gekennzeichnet ist, daß die Aufteilung in Verbindung mit Signalspeichermitteln durchgeführt wird, die einen ersten, einen zweiten, einen dritten und einen vierten Speicherbereich enthalten, wobei;

- (i) der erste Speicherbereich mehrere Bytes enthält, die jeweils ein aus dem ersten Zeichensatz ausgewähltes Zeichen darstellen,
- (ii) der zweite Speicherbereich mehrere Bytes enthält, die jeweils ein aus dem ersten Zeichensatz ausgewähltes Zeichen darstellen, wobei sich der Gesamtinhalt des zweiten Speicherbereiches von dem Gesamtinhalt des ersten Speicherbereiches unterscheidet,
- (iii) der dritte Speicherbereich Folgen enthält, die jeweils aus einem oder mehreren Bytes bestehen, wobei das Byte oder das erste Byte jeder Folge im ersten Speicherbereich enthalten ist, und
- (iv) der vierte Speicherbereich Folgen enthält, die jeweils aus einem oder mehreren im zweiten Speicher-

bereich enthaltenen Bytes bestehen.

Die im ersten Bereich gespeicherten Bytes stellen vorzugsweise Vokale dar, während die Bytes des zweiten Bereichs vorzugsweise Konsonanten darstellen. Überlappungen, z.B. der Buchstabe "y", sind möglich. Die Folgen im dritten Speicherbereich stellen vorzugsweise "Reime" dar, während die Folgen des vierten Speicherbereichs vorzugsweise Wortanfänge darstellen. Die Konzepte von Vokalen, Konsonanten, Reimen und Wortanfängen werden nachfolgend ausführlich erläutert.

Die Aufteilung umfaßt das Vergleichen von Unterfolgen des Eingabesignals mit Folgen, die im dritten und vierten Speicherbereich enthalten sind. Die Unterfolgen für den Vergleich werden durch Verwendung des ersten und des zweiten Speicherbereichs gebildet.

Die rückläufige Analyse erfordert, daß später vorkommende Unterfolgen vor früher vorkommenden Unterfolgen ausgewählt werden. Sobald eine Unterfolge ausgewählt wurde, stehen die darin enthaltenen Bytes nicht mehr zur Verfügung für die Auswahl oder Wiederauswahl, so daß sie eine früher vorkommende Unterfolge bilden. Diese Nicht-Verfügbarkeit begrenzt die Wahl für Bildung der Unterfolge, und deswegen bestimmt die vorausgehende Auswahl zumindest teilweise die spätere Auswahl der früheren Unterfolge.

Das erfindungsgemäße Verfahren ist besonders geeignet für die Verarbeitung einer Eingabefolge, die in Blöcke aufgeteilt ist, z.B. Blöcke, die Wörtern entsprechen, wobei ein Block in Segmenten analysiert wird, die am Ende beginnen und zum Beginn hin verarbeitet werden und wobei die Auswahl der Segmente vom Ende der restlichen nicht verarbeiteten Folge erfolgt.

Die Erfindung, die in den Patentansprüchen offenbart wird, umfaßt die Verfahren und eine Einrichtung zur Durchführung der Verfahren.

Die Datendarstellungen, z.B. Bytes, die beim erfindungsgemäßen Verfahren verwendet werden, nehmen jede beliebige Signalform an, die für Verwendung in Computerschaltungen geeignet sind. So können die Datendarstellungen durch Signale in Form von elektrischem Strom (Ampere), elektrischem Potential (Volt), magnetischem Feld, elektrischem Feld oder elektromagnetischer Strahlung gebildet werden. Zusätzlich können die Datendarstellungen gespeichert werden, einschließlich eines vorübergehenden Speicherns als Bestandteil der Verarbeitung, in einem geeigneten Speichermedium, z.B. als Grad und/oder Orientierung der Magnetisierung in einem magnetischen Medium.

Nachfolgend werden die theoretische Grundlage und einige vorteilhafte Ausführungen der Erfindung beschrieben. Bei den vorteilhaften Ausführungen werden die Eingangssignale in Blöcke aufgeteilt, die den einzelnen Wörtern des Textes entsprechen und die Erfindung arbeitet separat mit jedem Block, so daß der Prozess als "Wort-für-Wort"-Verarbeitung angesehen werden kann.

Nun ist es angebracht, die Anforderung zu wiederholen, daß es nicht erforderlich ist, für jeden der Blöcke eine Ausgabe zu erzeugen, da, wie oben erwähnt wurde, das gesamte System weitere Module umfaßt, die mit derartigen Ausfällen fertig werden.

Als eine Vorprüfung ist es sinnvoll, die theoretische Grundlage der Erfindung durch Betrachtung der Struktur von Wörtern in der englischen Sprache und durch Erklärung der

Strukturen von wenigen spezifischen Wörtern darzustellen. Diese Analyse verwendet den Unterschied, der gewöhnlich als "Vokale" und "Konsonanten" dargestellt wird. Für mechanische Verarbeitung ist es notwendig, zwei Zeichenverzeichnisse zu speichern. Eines dieser Verzeichnisse enthält die Zeichen, die als "Vokale" bezeichnet werden, und das andere Verzeichnis enthält diejenigen Zeichen, die als "Konsonanten" bezeichnet werden. Alle Zeichen sind vorzugsweise in einem oder dem anderen Verzeichnis enthalten, aber bei der vorteilhaften Ausführung sind die Datendarstellungen, die einem "Y" entsprechen, in beiden Verzeichnissen enthalten. Der Grund dafür ist, daß die gewöhnliche englische Rechtschreibung den Buchstaben "Y" manchmal als ein Vokal und manchmal als ein Konsonant verwendet. Somit enthält das erste Verzeichnis (der Vokale) a, e, i, o, u und y, während das zweite Verzeichnis die Konsonanten b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, w, x, y, z enthält. Die Tatsache, daß "Y" in beiden Verzeichnissen vorkommt, bedeutet, daß sich die Bedingung "nicht Vokal" von der Bedingung "Konsonant" unterscheidet.

Der primäre Zweck der Analyse ist es, einen Block von Datendarstellungen, d.h., ein Wort, in Reime und Wortanfänge zu teilen. Es ist wichtig zu begreifen, daß die Analyse verknüpfte Datenbanken verwendet, die Graphemäquivalente von Reimen und "Wortanfängen" enthält, die mit ihren Phonemäquivalenten verknüpft sind. Der Zweck der Analyse ist nicht nur, die Daten in die Reime und die Wortanfänge darstellende willkürliche Folgen zu teilen, sondern in Folgen, die in der Datenbank enthalten sind.

Ein Reim bezeichnet eine Folge von einem oder mehreren Zeichen, von denen jedes im Verzeichnis der Vokale enthalten ist, oder eine Folge, der eine zweite Folge von Zeichen folgt, die im Verzeichnis der Vokale nicht enthalten sind.

Eine alternative Erklärung dieser Anforderung ist, daß ein Reim aus einer ersten Folge besteht, der eine zweite Folge folgt, wobei alle in der ersten Folge enthaltenen Zeichen im Verzeichnis der Vokale enthalten sind und die erste Folge nicht leer sein darf und die zweite Folge vollständig aus Zeichen besteht, die im Verzeichnis der Vokale nicht enthalten sind mit dem Vorbehalt, daß die zweite Folge leer sein kann.

Ein Wortanfang ist eine Folge von Zeichen, von denen alle im Verzeichnis der Konsonanten enthalten sind.

Die Analyse erfordert, daß das Ende eines Wortes ein Reim sein soll. Es ist erlaubt, daß das Wort benachbarte Reime enthält, aber es ist nicht erlaubt, daß es benachbarte Wortanfänge enthält. Es wurde erwähnt, daß das Ende des Wortes ein Reim sein muß, aber es sollte erwähnt werden, daß der Anfang des Wortes entweder ein Reim oder ein Wortanfang sein kann; z.B. fängt das Wort "orange" mit einem Reim an, während das Wort "pear" mit einem Wortanfang beginnt.

Um die grundlegende Theorie der Erfindung zu erklären werden vier aus der englischen Sprache willkürlich ausgewählte Probewörter dargestellt und bezüglich ihrer Reime und Wortanfänge analysiert:

Erste Probe

CATS

Reim "ats"

Wortanfang "c"

Es ist zu erwarten, daß "ats" als ein Reim und "c" als ein Wortanfang aufgeführt werden. Deswegen wird jedes Ersetzen durch sein Phonemäquivalent "cats" in Phoneme umwandeln.

Es soll erwähnt werden, daß der Reim "ats" eine erste Folge, die aus dem einzelnen Vokal "a" besteht, und eine zweite Folge aufweist, die aus zwei Nicht-Vokalen, nämlich "t und "s" besteht.

Zweite Probe

STREET

Reim "eet"

Wortanfang "str"

In diesem Fall enthält die erste Folge des Reims zwei Buchstaben, nämlich "ee", während die zweite Folge durch ein einziges Nicht-Vokal "t" gebildet ist. Der Wortanfang besteht aus einer Folge von drei Konsonanten.

Der Wortanfang "str" und der Reim "eet" sollten beide in der Datenbank enthalten sein, so daß Phonemäquivalente vorhanden sind.

Dritte Probe

HIGH

Reim "igh"

Wortanfang "h"

Bei diesem Beispiel ist der Reim "igh" einer der beliebigen Klänge der englischen Sprache, aber die Datenbank kann eine korrekte Umwandlung in Phoneme gewährleisten.

Vierte Probe

HIGHSTREET

zweiter Reim "eet"

zweiter Wortanfang "str"

erster Reim "igh"

erster Wortanfang "h"

Es ist klar, daß das Wort "highstreet" eine Verbindung der

vorhergehenden zwei Beispiele ist und seine Analyse diesen zwei Beispielen sehr ähnlich ist. Es existiert jedoch eine wichtige Sonderanforderung, daß es notwendig ist, zu erkennen, daß eine Unterbrechung zwischen dem vierten und dem fünften Buchstaben vorhanden ist, um das Wort in "high" und "street" aufzuteilen. Diese Aufteilung wird durch den Inhalt der Datenbank erkannt. Somit ist die Konsonantenfolge "ghstr" kein Wortanfang in der englischen Sprache und ist deswegen in der Datenbank nicht enthalten, so daß sie nicht erkannt werden kann. Weiter ist auch die Folge "hstr" in der Datenbank nicht vorhanden. "str" ist jedoch im Englischen ein gewöhnlicher Wortanfang und sollte in der Datenbank vorhanden sein.

Deswegen kann "str" als ein Wortanfang erkannt werden und "str" ist der letzte Teil der Folge "ghstr". Sobald das Ende der Folge als ein Wortanfang erkannt wurde, wird der vordere Teil als Teil des vorhergehenden Reim identifiziert und das Wort "high" kann aufgeteilt werden, wie vorhin beschrieben wurde. Der Zweck dieses Beispiels ist darzustellen, daß die Aufteilung einer internen Konsonantenfolge manchmal wichtig ist und daß die Aufteilung durch Verwendung der Datenbank erreicht wird.

Nachdem nun die den erfindungsgemäßen Verfahren zugrundeliegende Theorie beschrieben wurde, ist es angebracht aufzuzeigen, wie sie durch Verwendung einer automatischen Computereinrichtung realisiert wird, die in der beliegenden diagrammatischen Zeichnung dargestellt ist.

Die Computereinrichtung arbeitet mit Folgen von Signalen, z. B., elektrischen Impulsen. Die kleinste Einheit der Computerverarbeitung ist eine Signalfolge, die einem einzigen Graphemen des Originaltextes entspricht. Zur Vereinfachung wird eine derartige Signalfolge als ein "Byte" bezeichnet,

wobei es unerheblich ist, wieviel Bit im Byte enthalten sind. Der Begriff "Byte" bezeichnet normalerweise eine Folge von 8 Bit. Da 8 Bit eine Zahl von 255 darstellen können, reicht diese Zahl aus, die meisten Alphabete unterzubringen. Das "Byte" enthält jedoch nicht unbedingt 8 Bit.

Die nachfolgend beschriebene Verarbeitung wird blockweise durchgeführt, wobei ein Block durch eine Folge von einem oder mehreren Bytes gebildet ist. Jeder Block entspricht einem individuellen Wort (oder einem potentiellen Wort, da es möglich ist, daß die Daten Blöcke enthalten, die nicht übersetzbar sind, so daß die Umwandlung ausfallen muß). Der Zweck des Verfahrens ist es, einen Eingabeblock, dessen Bytes Grapheme darstellen, in einen Ausgabeblock umzuwandeln, dessen Bytes Phoneme darstellen. Das Verfahren teilt den Eingabeblock in Unterfolgen auf, wandelt jede Unterfolge in eine Nachschlagetabelle um und bildet dann Ketten, um den Ausgabeblock zu erzeugen.

Der Betriebsmodus der Computereinrichtung weist zwei Arbeitsvorgänge auf. So weist er einen ersten Vorgang, der zwei Phasen umfaßt und zum Identifizieren von Bytefolgen verwendet wird, die den Reimen entsprechen. Der zweite Vorgang hat nur eine Phase und wird zum Identifizieren von Bytefolgen verwendet, die den Wortanfängen entsprechen.

Wie in der Zeichnung gezeigt ist, enthält die Computereinrichtung einen Eingabe-Puffer-Speicher 10, der Blöcke von der vorhergehenden Verarbeitung so lange hält, bis sie in der Lage sind, verarbeitet zu werden. Der Eingabe-Puffer-Speicher 10 ist mit einem Datenspeicher 11 verbunden und stellt dem Datenspeicher 11 auf Anforderung einzelne Blöcke zur Verfügung.

Ein wichtiges Teil der Computereinrichtung ist der Speicher

12. Dieser enthält Programmbefehle und auch die Datenbanken und Verzeichnisse, die zur Durchführung der Verarbeitung erforderlich sind. Wie im nachfolgenden Text ausführlich erläutert wird, ist der Speicher 12 in mehrere Funktionsbereiche unterteilt.

Die Datenverarbeitungseinrichtung umfaßt auch einen Arbeitsspeicher 14, der erforderlich ist, Untersätze von Bytes zu halten, die vom Datenspeicher 11 erhalten werden, zur Verarbeitung und zum Vergleich mit Bytefolgen, die in im Speicher 12 enthaltenen Datenbanken gehalten werden. Einzelne Bytes, d.h. Signalfolgen, die einzelnen Graphemen entsprechen, werden vom Eingabe-Puffer-Speicher 10 in den Arbeitsspeicher 14 über einen Kontrollspeicher 13 übertragen, der eine Kapazität von einem Byte aufweist. Vor der Übertragung in den Arbeitsspeicher 14 wird das Byte im Kontrollspeicher 13 mit Verzeichnissen verglichen, die im Datenspeicher 12 enthalten sind.

Nach dem erfolgreichen Vergleich mit den im Arbeitsspeicher 12 enthaltenen Datenfeldern werden die Folgen von dem Arbeitsspeicher 12 in den Ausgabespeicher 15 übertragen. Für den Fall, daß die Übereinstimmung nicht stattfindet, enthält die Einrichtung Mittel zur Rückführung eines Bytes vom Arbeitsspeicher 14 in den Datenspeicher 11.

Zusätzlich zu anderen Bereichen, z. B. für Programmbefehle, weist der Speicher 12 vier Hauptspeicherbereiche auf. Diese Bereiche werden nun erläutert.

Zunächst weist der Speicher Bereiche für zwei verschiedene Verzeichnisse von Bytes auf. Dies sind ein erster Speicherbereich 12.1, der ein Verzeichnis von Bytes enthält, die den Vokalen entsprechen, und ein zweiter Speicherbereich 12.2, der ein Verzeichnis von Bytes enthält, die den Konso-

nanten entsprechen. (Die Vokale sowie die Konsonanten wurden bereits in dieser Beschreibung erläutert).

Der Speicher 12 enthält auch zwei Speicherbereiche, die zwei verschiedene und wichtige verknüpfte Datenbanken bilden. Die erste ist die "Reim"-Datenbank 12.3, die weiter in Bereiche unterteilt ist, die mit 12.31, 12.32, 12.33 usw. bezeichnet sind. Jeder Bereich hat einen Eingabeteil, der Bytefolgen enthält, die den Reimen in Graphemen entsprechen, wobei, wie in der Zeichnung dargestellt ist, 12.31 "ATS" enthält, 12.32 "EET" enthält, 12.33 "IGH" enthält, sowie mehrere Teile, die in der Zeichnung nicht dargestellt sind.

Der Speicher 12 enthält auch einen zweiten Hauptbereich 12.4, der Bytefolgen enthält, die den Wortanfängen entsprechen. Wie bei den Reimen ist die Wortanfang-Datenbank 12.4 auch in mehrere Bereiche unterteilt. So zum Beispiel umfaßt sie 12.41, die "C" enthält, 12.42, die "STR" enthält und 12.43, die "H" enthält.

Jeder der Eingabeteile (von 12.3 und 12.4) ist mit einem Ausgabeteil verknüpft, der eine Folge von Bytes enthält, die dem Inhalt seines Eingabeteiles entsprechen.

Es wurde bereits erwähnt, daß das Arbeitsverfahren zwei verschiedene Vorgänge umfaßt. Der erste Vorgang benutzt die Speicherbereiche 12.1 und 12.3, während der zweite Vorgang die Speicherbereiche 12.2 und 12.4 benutzt. Es wird betont, daß die Datenbankbereiche, die derzeit verwendet werden, vollständig durch den laufenden Vorgang definiert werden. Die Vorgänge werden abwechselnd verwendet und der Vorgang Nr. 1 wird als erstes verwendet.

TYPISCHES BEISPIEL

Analyse des Wortes "HIGHSTREET"

Es wird bemerkt, daß dieses typische Beispiel das Wort betrifft, das als die vierte Probe in der vorhergehenden Beschreibung ausgewählt wurde. Deswegen wurden seine Reime und Wortanfänge bereits festgelegt und das typische Beispiel erklärt, wie diese durch mechanische Berechnung erreicht wurden.

Die Analyse beginnt, wenn der Eingabe-Puffer-Speicher 10 die dem Wort "HIGHSTREET" entsprechende Bytefolge in den Datenspeicher 12 überträgt. Somit sehen die Inhalte der wichtigen Speicher zu Beginn des Prozesses wie folgt aus:

<u>SPEICHER</u>	<u>INHALT</u>
11	HIGHSTREET
13	--
14	--
15	--

(Das Zeichen "--" bedeutet, daß der entsprechende Speicher leer ist).

Die Analyse beginnt mit dem ersten Vorgang, weil eine Analyse immer mit dem ersten Vorgang beginnt. Wie oben erwähnt wurde, benutzt der erste Vorgang die Speicherbereiche 12.1 und 12.3. Der erste Vorgang hat zwei Phasen, während derer Bytes vom Datenspeicher 11 in den Arbeitsspeicher 14 über den Kontrollspeicher 13 übertragen werden. Die erste Phase wird so lange fortgesetzt, bis im Speicherbereich 12.1 keine Bytes gefunden werden.

Der Vorgang läuft rückwärts ab, was bedeutet, daß er am Ende des Wortes beginnt, so daß als erstes das "T" über-

tragen wird, das im Bereich 12.1 nicht enthalten ist. Als zweites wird das "E" übertragen, das im Bereich 12.1 enthalten ist, wodurch die zweite Phase des ersten Vorgangs eingeleitet wird. Dies wird so lange fortgesetzt, bis das Byte im Arbeitsspeicher 14 mit 12.1 übereinstimmt, so daß das zweite "E" übertragen wird, aber die Überprüfung ausfällt, wenn das nächste Byte "R" durchläuft. In diesem Stadium sieht der Zustand der Speicher wie folgt aus:

<u>SPEICHER</u>	<u>INHALT</u>
11	HIGHST
13	R
14	EET
15	- - - -

Die Inhalte des Arbeitsspeichers 14 werden zum Zugriff auf den Speicherbereich 12.3 verwendet, wobei eine Übereinstimmung im Bereich 12.32 stattfindet. Somit hat die Übereinstimmung stattgefunden und der Inhalt des Arbeitsspeichers 14, nämlich "EET" wird in einen Bereich des Ausgangsspeichers 15 übertragen, so daß der Zustand der Speicher folgendermaßen aussieht.

<u>SPEICHER</u>	<u>INHALT</u>
11	HIGHST
13	R
14	- - -
15	EET

Es wird gemerkt, daß das erste Reim mechanisch gefunden wurde.

Wie oben erwähnt wurde, wurde durch die Nicht-Übereinstimmung von "R" im Kontrollspeicher 13 die erste Durchführung des ersten Vorgangs beendet. Die Analyse wird fort-

gesetzt, aber nun wird der zweite Vorgang verwendet, weil die zwei Vorgänge immer abwechselnd durchgeführt werden. Der zweite Vorgang verwendet die Speicherbereiche 12.2 und 12.4. Das dem "R" entsprechende Byte im Kontrollspeicher 13 stimmt nun überein, da nun der Bereich 12.2 benutzt wird und dieses Byte darin enthalten ist. Deswegen wird "R" in den Arbeitsspeicher 14 übertragen und der zweite Vorgang wird so lange fortgesetzt, bis eine Übereinstimmung des Bytes im Kontrollspeicher 13 stattfindet. Somit werden alle Buchstaben "T", "S", "H" und "G" über den Kontrollspeicher 13 übertragen. In diesem Augenblick kommt das dem "I" entsprechende Byte in den Kontrollspeicher 13 und die Überprüfung entfällt, da das dem "I" entsprechende Byte nicht im Speicherbereich 12.2 enthalten ist. Da die Überprüfung entfällt, wird die Durchführung des zweiten Vorgangs beendet. Die Inhalte der Speicher sehen folgendermaßen aus.

<u>SPEICHER</u>	<u>INHALT</u>
11	"H"
13	"I"
14	"GHSTR"
15	"EET"

Der zweite Vorgang wird versuchen, den Inhalt des Arbeitsspeichers 14 mit der in 12.4 enthaltenen Datenbank zu vergleichen; es wird jedoch keine Übereinstimmung erreicht. Deswegen wird der zweite Vorgang mit seinem Bedarfsteil fortgesetzt, wobei die Bytes über den Kontrollspeicher 13 zurück in den Datenspeicher 11 übertragen werden. Bei jeder Übertragung wird versucht, den Inhalt des Arbeitsspeichers 14 in den Speicherbereich 12.4 zu bringen. Eine Übereinstimmung wird erreicht, wenn die Buchstaben G und H zurückgeführt wurden, weil die dem "STR" äquivalente Folge im Bereich 12.42 enthalten ist. Nachdem eine Übereinstimmung erreicht wurde, wird der Inhalt des Arbeitsspeichers in

19: 12:00

einen Bereich des Ausgabespeichers 15 ausgegeben. Zu diesem Zeitpunkt sieht der Inhalt der Speicher folgendermaßen aus.

<u>SPEICHER</u>	<u>INHALT</u>
11	"HIG"
13	"H"
14	- -
15	"STR" und "EET"

Der zweite Vorgang wurde durch Erreichen der Übereinstimmung beendet, so daß die Analyse nun zum ersten Vorgang zurückkehrt, insbesondere zur ersten Phase des ersten Vorgangs. Auf diese Weise werden die Buchstaben "H" und "G" in den Arbeitsspeicher 14 übertragen und die erste Phase endet. Die zweite Phase läßt "I" durch und wird beendet, wenn "H" in den Kontrollspeicher 13 übertragen wurde. In diesem Stadium weisen die Speicher die folgenden Inhalte auf. :

<u>SPEICHER</u>	<u>INHALT</u>
11	- -
13	"H"
14	"IGH"
15	"STR" und "EET".

Der erste Vorgang versucht nun, den Inhalt des Arbeitsspeichers 14 mit der Datenbank im Speicherbereich 12.3 zu vergleichen, wobei eine Übereinstimmung im Bereich 12.33 erreicht wird. Deswegen wird der Inhalt des Arbeitsspeichers 14 in einen Bereich des Ausgabespeichers 15 übertragen.

Die Analyse wird nun mit dem zweiten Vorgang fortgesetzt und der Buchstabe "H" (im Kontrollspeicher 13) wird in den Speicherbereich 12.2 gebracht (merke, daß dieser Bereich nun verwendet wird, da die Analyse zum zweiten Vorgang zurückgekehrt ist). Die Analyse kann jetzt beendet werden,

da der Datenspeicher 11 keine weiteren Bytes zur Übertragung beinhaltet und der Inhalt des Arbeitsspeichers, nämlich "H", im Bereich 12.43 des Hauptspeichers 12 gefunden wird. Somit wird "H" zum Ausgabespeicher 15 übertragen, der die korrekten vier Folgen enthält, die durch mechanische Analyse gefunden wurden.

Nachdem die erforderlichen Ausgabefolgen in die richtige Lage gebracht wurden, ist es lediglich notwendig, sie auf Grund der Tatsache umzuwandeln, daß die Speicherbereiche 12.3 und 12.4 verknüpfte Datenbanken sind. Jeder Bereich enthält nicht nur die Folgen, die nun im Ausgabespeicher enthalten sind, sondern sind mit jedem Bereich Ausgabebereiche verknüpft, die Folgen enthalten, die den geeigneten Phonemen entsprechen. Deswegen wird jede Folge im Ausgabespeicher zum Zugriff auf ihren geeigneten Bereich verwendet und erzeugt somit die erforderliche Ausgabe. Der Endschrift verwendet lediglich eine Nachschlagetabelle und dies ist deswegen möglich, weil die wichtige Analyse beendet wurde.

Wie oben erwähnt wurde, dienen die identifizierten Folgen als Zugriff auf die verknüpfte Datenbank und in einem einfachen System existiert für jede Zugriffsfolge eine Ausgabefolge. Die Aussprache hängt jedoch manchmal vom Kontext ab und eine bessere Umwandlung kann durch Verwendung mehrerer Ausgaben für mindestens einige der Zugriffsfolgen erreicht werden. Die Auswahl des geeigneten Ausgabestroms hängt von der Analyse des Kontextes des Zugriffsstroms ab, d.h., von der Berücksichtigung der Position im Wort oder davon, was folgt oder was fortgesetzt wird. Diese weitere Komplikation beeinflusst die Erfindung nicht, die sich lediglich mit der Aufteilung in geeignete Bereiche befaßt. Sie kompliziert lediglich den Nachschlageprozess.

Wie oben erläutert wurde, ist die Erfindung nicht unbedingt erforderlich, um eine Ausgabe zu erzeugen, weil in einem Fehlerfall das ganze System ein Standard-Verfahren enthält, das z. B. ein Phonem zur Verfügung stellt, das jedem Graphem äquivalent ist. Um die Beschreibung des Verfahrens zu ergänzen ist es erwünscht, eine kurze Erläuterung des Umstands zu geben, bei dem der Fehlerfall eintritt und die Verwendung des Standard-Verfahrens erforderlich ist.

Fehlermodus 1.

Der erste Fehlermodus tritt auf, wenn der Inhalt des Datenspeichers kein Vokal enthält, was darauf schließen läßt, daß der Inhalt kein Wort darstellt. Wie immer beginnt die Analyse mit dem ersten Vorgang und insbesondere mit der ersten Phase des ersten Vorgangs und wird so lange fortgesetzt werden, bis keine Übereinstimmung mit dem ersten Verzeichnis 12.1 stattfindet. Da die Folge und der Datenspeicher 11 keine Übereinstimmung enthalten, wird die erste Phase bis zum Anfang des Wortes fortgesetzt und dies bedeutet, daß ein Fehler auftritt.

Zweiter Fehlermodus

Dieser Fehler tritt auf, wenn:

- (i) der zweite Vorgang läuft;
- (ii) der Anfang des Wortes erreicht wird und;
- (iii) keine Übereinstimmung für den Inhalt des Arbeitsspeichers 14 in der Datenbank 12.4 existiert.

Dies steht im Gegensatz zum Übereinstimmungsfehler in der Mitte des Wortes, der darauf schließen läßt, daß im Kontrollspeicher 13 ein Vokal enthalten ist. Ein Fehler in diesem Stadium ermöglicht eine Rückführung von Bytes für eine spätere Analyse durch den ersten Vorgang und wird, zumindest in diesem Punkt der Analyse, nicht als Fehler angesehen. Wenn der Anfang des Wortes erreicht ist, besteht

keine Möglichkeit einer weiteren Analyse, so daß die Analyse entfallen muß.

Dritter Fehlermodus

Der dritte Fehlermodus tritt auf, wenn der erste Vorgang läuft und wenn es nicht möglich ist, die Inhalte des Arbeitsspeichers 14 mit einer in der Datenbank 12.3 enthaltenen Folge in Übereinstimmung zu bringen. Unter diesen Umständen wird der erste Vorgang Bytes zurück zum Kontrollspeicher 13 und zum Datenspeicher 11 übertragen und diese Übertragung kann so lange fortgesetzt werden, bis der Arbeitsspeicher 14 leer ist und die Analyse somit entfällt.

Im zweiten Fehlermodus wurde erläutert, daß der zweite Ablauf in der Lage ist, Bytes zur späteren Analyse durch den zweiten Vorgang an den Eingang zurück zu bringen. Die übertragenen Bytes müssen jedoch zu einem bestimmten Zeitpunkt, d.h., während der nächsten Durchführung des ersten Vorgangs übereinstimmen. Der dritte Fehlermodus entspricht dem Fall, in dem es nicht möglich ist, die spätere Übereinstimmung zu erreichen.

Das erfindungsgemäße Verfahren führt somit eine Aufteilung einer Datenfolge in Segmente durch, die durch Verwendung von Nachschlagetabellen umgewandelt werden können. Es ist nicht notwendig, daß die Aufteilung in jedem Fall zu erfolgen hat, aber unter der Voraussetzung, daß hochwertige Datenbanken verwendet werden, das Verfahren sehr schnell arbeiten und die Leistung des gesamten Systems erhöhen wird, das andere Module aufweist, die für die Umwandlung eines Textes in eine Sprache erforderlich sind.

Patentansprüche

1. Verfahren zur Verarbeitung eines Eingangssignals, das aus einer Folge von Bytes besteht, die jeweils einem Zeichen aus einem ersten Zeichensatz entsprechen, um Unterfolgen für eine Umwandlung in ein Ausgangssignal zu identifizieren, das eine Folge von Zeichen darstellt, die aus einem sich von dem ersten Zeichensatz unterscheidenden zweiten Zeichensatz ausgewählt werden, wobei das Verfahren das Eingangssignal durch eine rückläufige Analyse in Unterfolgen aufteilt,
dadurch gekennzeichnet, daß
die Aufteilung in Verbindung mit einer Datenbank in Form von Signalen durchgeführt wird, die in einem ersten, zweiten, dritten und vierten Speicherbereich gespeichert sind, wobei:
 - (i) der erste Speicherbereich (12.1) mehrere Bytes enthält, die jeweils ein aus dem ersten Zeichensatz ausgewähltes Zeichen darstellen,
 - (ii) der zweite Speicherbereich (12.2) mehrere Bytes enthält, die jeweils ein aus dem ersten Zeichensatz ausgewähltes Zeichen darstellen, wobei sich der Gesamtinhalt des zweiten Speicherbereiches von dem Gesamtinhalt des ersten Speicherbereiches unterscheidet,
 - (iii) der dritte Speicherbereich (12.3) Folgen enthält, die jeweils aus einem oder mehreren Bytes bestehen, wobei das Byte oder das erste Byte jeder Folge im ersten Speicherbereich enthalten ist, und
 - (iv) der vierte Speicherbereich (12.4) Folgen enthält, die jeweils aus einem oder mehreren im zweiten Speicherbereich enthaltenen Bytes bestehen,die Aufteilung das Vergleichen von Unterfolgen (12.3, 12.4, 14) des Eingangssignals mit Folgen aufweist, die in

dem dritten und vierten Bereich der Signalspeichereinrichtung enthalten sind, und Auswählen von später vorkommenden Unterfolgen vor früher vorkommenden Unterfolgen, wobei die vorherige Auswahl einer späteren Unterfolge zumindest teilweise die Auswahl einer früheren Unterfolge definiert,

die Unterfolgen für den Vergleich durch Vergleichen (12.1, 12.2, 13) von Bytes des Eingangssignals mit den Inhalten des ersten und des zweiten Speicherbereiches gebildet werden, um Unterfolgen, die mit einem im ersten Speicherbereich enthaltenen Byte anfangen oder aus diesem bestehen, und andere Folgen zu bilden, die vollständig aus im zweiten Speicherbereich enthaltenen Bytes bestehen.

2. Verfahren nach Anspruch 1, bei dem das Eingangssignal in Blöcke aufgeteilt wird und die Verarbeitung von zumindest einigen dieser Blöcke aufweist:
 - (a) Identifizieren einer inneren Folge von aufeinanderfolgenden Bytes, die jeweils im zweiten Speicherbereich enthalten sind, wobei die Folge unmittelbar an ein im ersten Speicherbereich enthaltenes vorangehendes Byte anschließt, und unmittelbar einem im ersten Speicherbereich enthaltenen nachfolgenden Byte vorausgeht,
 - (b) Identifizieren der Folge mit dem längsten Ende aus der inneren Folge mit einer Folge, die im vierten Speicherbereich enthalten ist,
 - (c) Definieren eines Anfangsteils der inneren Folge als den nach der in (b) definierten Abtrennung der Endfolge verbleibenden Rest,
 - (d) Identifizieren einer Folge aus einem oder mehreren aufeinanderfolgenden Bytes, die jeweils im ersten Speicherbereich enthalten sind, wobei die Folge das in (a) identifizierte vorangehende Byte enthält, und

- (e) Verbinden des in (c) identifizierten Anfangsteils mit der in (d) identifizierten Folge, um eine im dritten Speicherbereich gespeicherte Folge zu erzeugen.
3. Verfahren nach Anspruch 1 oder 2, bei dem jede im dritten Speicherbereich enthaltene Folge aus einer Primärfolge und einer nachfolgenden Sekundärfolge besteht, wobei die Primärfolge aus im ersten Speicherbereich enthaltenen Bytes besteht und die zweite Folge entweder leer ist oder aus im zweiten Speicherbereich enthaltenen Bytes besteht.
4. Verfahren zur Umwandlung eines Eingangssignals, das eine Folge von aus dem ersten Zeichensatz ausgewählten Zeichen darstellt, in ein äquivalentes Signal, das eine Folge von aus dem zweiten Zeichensatz ausgewählten Zeichen darstellt, mit Identifizieren von Unterfolgen durch ein Verfahren nach einem der vorangehenden Ansprüche, und Umwandeln der Unterfolgen mittels einer verbundenen Datenbank, die Eingangsabschnitte mit jeweils einer der Unterfolgen enthält, wobei jeder Eingangsabschnitt mit einem Ausgangsabschnitt verbunden ist, der die zum Inhalt des Eingangsabschnitts äquivalente Ausgabe enthält.
5. Verfahren nach Anspruch 4, bei dem das Eingangssignal in Eingangsblöcke aufgeteilt wird und bei dem jeder Block für sich umgewandelt wird, wobei zumindest einige der Blöcke als Ganzes ohne Unterteilung umgewandelt werden und zumindest einige der Blöcke durch ein Verfahren nach Anspruch 4 umgewandelt werden.
6. Zweiteilige Datenbank zum Einfügen in eine Sprachmaschine zur Durchführung eines Verfahrens nach Anspruch 4 oder 5, wobei die Datenbank als in Signalspeichereinrichtungen gespeicherte Signale ausgebildet ist und aufweist:
 - (i) einen ersten Speicherbereich (12.1), der mehrere Bytes enthält, die jeweils ein aus dem ersten Zeichensatz ausgewähltes Zeichen darstellen,

- (ii) einen zweiten Speicherbereich (12.2), der mehrere Bytes enthält, die jeweils ein aus dem ersten Zeichensatz ausgewähltes Zeichen darstellen, wobei sich der Gesamtinhalt des zweiten Speicherbereiches von dem Gesamtinhalt des ersten Speicherbereiches unterscheidet,
 - (iii) einen dritten Speicherbereich (12.3), der aus einem oder mehreren Bytes bestehende Zeichen enthält, wobei das Byte oder das erste Byte jeder Folge im ersten Speicherbereich enthalten ist, jede im dritten Speicherbereich (12.3) enthaltene Folge mit einem Ausgangsregister verbunden ist, das eine Folge aus einem oder mehreren Bytes enthält, die jeweils ein Zeichen des zweiten Zeichensatzes darstellen, und das Zeichen im Ausgangsregister eine Umwandlung der im dritten Speicherbereich (12.3) enthaltenen verbundenen Folge darstellt, und
 - (iv) einen vierten Speicherbereich (12.4), der aus einem oder mehreren, im zweiten Speicherbereich enthaltenen Bytes bestehende Folgen enthält, die mit einem Ausgangsregister verbunden sind, das eine Folge aus einem oder mehreren Bytes enthält, die jeweils ein Zeichen des zweiten Zeichensatzes darstellen, wobei die Folge im Ausgangsregister eine Umwandlung der im vierten Speicherbereich (12.4) enthaltenen verbundenen Folge darstellt.
7. Zweiteilige Datenbank nach Anspruch 6, bei der jede im dritten Speicherbereich enthaltene Folge aus einer Primärfolge und einer nachfolgenden Sekundärfolge besteht, wobei die Primärfolge aus im ersten Speicherbereich enthaltenen Bytes besteht und die Sekundärfolge entweder leer ist oder aus im zweiten Speicherbereich enthaltenen Bytes besteht.
8. Sprachmaschine, die eine zweiteilige Datenbank nach Anspruch 6 oder 7 enthält.

EP 0 691 023

